

IN THE CLAIMS:

This listing of claims will replace all prior versions, and listings, of claims in the present application:

1-24. (Canceled)

25. (Currently amended) A method comprising:

maintaining an active map of information indicating in-use blocks and free blocks of an active file system in a storage system;

maintaining a set of snapshots in the storage system, each snapshot representing a state of said active file system at a particular point in time, each snapshot having a corresponding active map indicating in-use blocks and free blocks of the active file system for a point in time at which said snapshot was generated;

computing a summary map as a logical OR of the active maps of at least two of said snapshots; and

using the summary map to make write allocation decisions in the storage system, including using the summary map to avoid overwriting blocks used by a snapshot.

26. (Canceled)

27. (Previously presented) A method as in claim 25, wherein said summary map is computed using an inclusive OR operation.

28. (Original) A method as in claim 25, wherein
said set of snapshots includes at least two said snapshots; and
said computing includes performing a bitwise logical operation on at least two
said copies of earlier active maps included in said set of snapshots.

29. (Previously presented) A method as in claim 25, wherein using the
summary map to make write allocation decisions in the storage system comprises:
making write allocation decisions based on both a current active map of the
active file system and said summary map.

30. (Previously presented) A method as in claim 25, wherein using the
summary map to make write allocation decisions in the storage system comprises:
computing a combination of a current active map and said summary map; and
making write allocation decisions based on a result of said computing.

31. (Previously presented) A method as in claim 25, further including, for a
selected portion of said summary map
identifying a set of snapshots created since a recent update of said selected
portion; and
updating said selected portion based on only a most recent one of said
snapshots.

32. (Currently amended) A method comprising:

maintaining an active map of information indicating in-use blocks and free blocks of an active file system;

maintaining a set of snapshots, each snapshot representing a state of said active file system at a particular point in time, each snapshot having a corresponding active map indicating in-use blocks and free blocks of the active file system for a point in time at which said snapshot was generated;

maintaining a summary map based on an active map of at least one of said snapshots;

making a write allocation decision based on the summary map, including using the summary map to avoid overwriting blocks used by a snapshot;

receiving a request to delete a particular snapshot; and

deleting said particular snapshot, wherein said deleting involves, for a block used by said particular snapshot, indicating said block is free in said summary map depending on a snapshot just prior to said particular snapshot and a snapshot just after said particular snapshot.

33. (Previously presented) A method as in claim 32, wherein said indicating frees said block only when both

said block is unused by said snapshot just prior to said particular snapshot; and

said block is unused by said snapshot just after said particular snapshot.

34. (Previously presented) A method as in claim 32, wherein said snapshot just after said particular snapshot corresponds to an active file system.

35. (Currently amended) A method comprising:
maintaining an active map of information indicating in-use and free blocks associated with a file system;

maintaining a set of snapshots, each snapshot representing a state of said file system at a particular point in time;

maintaining a summary map computed as a logical OR of active maps included in at least two of said snapshots;

selecting a set of blocks maintained by said file system for which to perform a write allocation operation, including using the summary map to avoid overwriting blocks used by a snapshot;

updating only a portion of said summary map corresponding to said set of blocks, in response to said selecting; and

performing said write allocation operation in response to said updated summary map.

36–39. (Canceled)

40. (Previously presented) A method as in claim 35, wherein said summary map is computed using an inclusive OR operation.

41. (Previously presented) A method as in claim 32, wherein said summary map represents a logical OR of at least two copies of an earlier active map included in at least two of said snapshots.

42. (Previously presented) A method as in claim 41, wherein said logical union is an inclusive OR operation.

43. (Currently amended) A method comprising:

maintaining a plurality of persistent point-in-time images of a file system, each persistent point-in-time image representing a state of said file system at a particular point in time, each persistent point-in-time image having associated therewith a separate map indicating in-use blocks and free blocks of the file system at the corresponding point in time; ~~and~~

generating a summary map as a logical OR of said maps associated with at least two of said persistent point-in-time images; and

making write allocation decisions based on said summary map and a map indicating in-use blocks and free blocks associated with a current state of the file system, including using the summary map to avoid overwriting blocks used by a snapshot.

44. (Canceled)

45. (Previously presented) A method as in claim 44, wherein

said summary map is computed using an inclusive OR operation.

46. (Previously presented) A method as in claim 43, wherein said generating includes performing a bitwise logical operation on at least two of said maps associated with the plurality of persistent point-in-time images.

47. (Canceled)

48. (Previously presented) A method as in claim 43, further including:
determining a combination of said summary map and a map indicating in-use blocks and free blocks associated with a current state of the file system; and
making write allocation decisions based on a result of said determining.

49-50. (Canceled)

51. (Currently amended) A method comprising:
maintaining a plurality of snapshots of a structured set of data in a data storage system, each snapshot representing a state of said structured set of data at a particular point in time, each snapshot having associated therewith a separate active map indicating in-use blocks and free blocks of the structured set of data at the corresponding point in time;

generating a summary map which represents a summary of at least two or more of said active maps for different points in time, by using a logical OR of ~~the said at least~~ two ~~or more~~ of said active maps; and

making write allocation decisions relating to the structured set of data in the data storage system, based on the summary map, including using the summary map to avoid overwriting blocks used by a snapshot.

52. (Previously presented) A method as in claim 51, wherein the data storage system comprises a file server and the structured set of data is a file system.

53. (Previously presented) A method as in claim 51, wherein the logical OR is an inclusive OR operation.

54. (Previously presented) A method as in claim 51, wherein said generating comprises performing a bitwise logical operation on at least two of said active maps.

55. (Previously presented) A method as in claim 51, wherein said making write allocation decisions is further based on an active map indicating in-use blocks and free blocks associated with a current state of the structured set of data.